Monterey Security Enhanced Architecture Project

Cynthia Irvine, David Shifflett, Paul Clark, Timothy Levin, George Dinolt
Naval Postgraduate School

Abstract

This research project has produced an innovative architecture and corrresponding engineering prototype consisting of trusted security services and integrated operating system mechanisms for the protection of distributed multi-domain computing environments from malicious code and other attacks. These security services and mechanisms extend and interoperate with existing workstations, applications and open source operating systems, providing new capabilities for composing secure distributed systems using commercial off-the-shelf (COTS) components. The latter construct results from the realization that unless a secure system offers users comfortable and familiar interfaces for handling routine information, the secure system will fail due to lack of user acceptability. This work was supported in part by the MYSEA project of the DARPA/ATO CHATS program.

1. Introduction

The purpose of the Monterey Security Enhanced Architecture (*MYSEA*, pronounced, `my-SEE-ah`) project is to provide a trusted distributed operating environment for enforcing multi-domain security policies, which supports unmodified COTS productivity applications. The architecture encompasses a combination of many low-assurance commercial components and relatively few specialized (e.g., high-assurance) multi-domain components. This arrangement permits the ongoing DoD and U.S. Government investment in commodity personal computer (PC) operating systems and applications to be integrated into an environment where enforcement of critical security policies is assigned to more trusted elements. Assurance is derived from the application of high assurance system design and development methods to the trusted elements as well as to the overall architecture.

The trusted computing base (TCB) for MYSEA is called the Monterey Secure Architecture Operating System (*MYSEOS*, pronounced, `my-SEE-ose`). In the MYSEA prototype we have constructed, MYSEOS is based upon a security-enhanced version of the OpenBSD operating system. We selected OpenBSD as the basis because the secure code reviews utilized by OpenBSD [11] provide a level of confidence in correct implementation that is not available in other open source operating systems. However, the operating system modifications we have defined are modular and conceptually simple enough that they could be accomplished on a variety of open source platforms (e.g., Linux). While the architecture can support higher assurance TCB components, the use of unevaluated and possibly unevaluatable operating systems in this research demonstration prototype could not be expected to achieve the assurance required for the secure management of information having a range of sensitivity levels [10].

We provide a mechanism for vertical integration of application security requirements with underlying security services, applying an existing Quality of Security Service model and framework [7] to the integrated security structure. Additionally, the MYSEA system supports a secure unforgeable bidirectional connection, called a *trusted path*, between the user and the trusted elements of the system.

Several aspects of this research provide innovative advances in the state of the art for protecting multiple domains of information and for the management of security policies and security services in support of critical applications (for a discussion of related work, see [8]). Ultimately, the commercial proliferation of these innovations will be available for direct consumption by the DoD for use by operational forces as well as for critical national information infrastructure systems. Specific innovations that are suitable for immediate technical transfer to commercial products are:

- A distributed architecture for isolating trusted components in support of commercial and open source applications. The innovative use of add-on components in commercial client-server systems can potentially magnify the impact of trusted open source systems.
- An open source trusted path mechanism for assured and unambiguous user communication with the trusted computing base.
- Techniques for vertical integration of security policy control functions with underlying security services in a Quality of Security Service

framework.

- Single sign-on for access to a community of distributed multi-domain policy servers. Once a user has authenticated to MYSEOS, application sessions may be transferred to any confederated MYSEA Server.

2. Monterey Security Enhanced Architecture (MYSEA)

MYSEA is a distributed client-server architecture featuring a combination of (relatively few) specialized policy enforcing components and multiple open source and commercial off-the-shelf components. The major physical components of the architecture are illustrated in Figure 1:

- Security enhanced servers which provide the locus for security policy enforcement and host various open source or commercial application protocol servers, and
- Security enhanced workstations that consist of commercial-class PCs executing popular commercial software products, along with Trusted Path Extensions that provide trustworthy policy support mechanisms and thus permit server-enforced security policy to be distributed across the network.

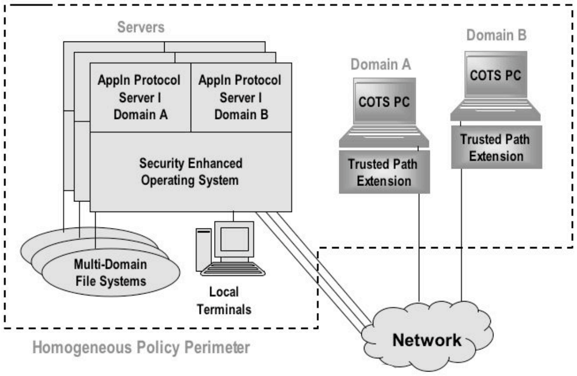
Servers
Appln Protocol Server I Domain A
Appln Protocol Server I Domain B
Security Enhanced Operating System
Domain A
COTS PC
Trusted Path Extension
Domain B
COTS PC
Trusted Path Extension
Multi-Domain File Systems
Local Terminals
Network
Homogeneous Policy Perimeter

Figure 1. Monterey Security Enhanced Architecture (MYSEA)

The MYSEA Server enforces the security policy and controls access to information. At its heart is a security-enhanced version of the OpenBSD operating system (MYSEOS). Application protocol servers run on the trusted server and provide services and interfaces to shared resources. When MYSEOS is combined with untrusted, but policy constrained (and, in some instances, policy aware) application protocol servers, the result is the MYSEA Server. Each MYSEA workstation is a PC equipped with a Trusted Path Extension device that provides MYSEA policy support at the workstation. The MYSEA Server(s) and the Trusted Path Extension(s) are the only components directly connected to the physical network. Multiple MYSEA Servers provide scalability within the desired security policy perimeter.

2.1 MYSEA concept of operation

Using the Trusted Path Extension at the PC, users log on to the MYSEA system by way of a trusted path, establishing an identity for audit and access control purposes, and then establish session properties such as current sensitivity level. Subsequently, the user can log on to the native client OS at the PC and use standard commercial client software (e.g., web browser or e-mail program) to access applications supported by the MYSEA Server, or use any applications supported by the local PC. From the PC the user can access any domain of server data allowed by the security policy (for example, reading domains of data that are lower in sensitivity that the negotiated session level) as well as access local data. By again invoking the trusted path, the user can request to modify session security attributes, such as "session level." During such negotiations, the Trusted Path Extension will ensure that client access to the network is blocked.

2.2 MYSEA components

The MYSEA system consists of the following hierarchy of components, which are described below.

- MYSEA Server
    - Policy-aware application protocol servers
    - MYSEOS
        - Trusted path services
        - Security Support Services
        - Secure session services
        - Quality of security services
        - Cryptographic services
        - Multi-domain open source kernel (MLS-enhanced OpenBSD)
- MYSEA Workstation
    - Trusted Path Extension
    - COTS PC, including unmodified:
        - Operating system
        - User interface
        - Applications
        - Network connections

2.3 MYSEA server

Each MYSEA Server consists of MYSEOS, which enforces critical security policy, and assorted untrusted application server instances (e.g. one per security domain per user). The actions of the application servers are constrained by the policy enforcement mechanisms of MYSEOS. The application servers are functionally equivalent in terms of overall application-level protocol support to a COTS application server for the particular protocol provided. Thus, each application server is compatible with existing COTS client packages. Additionally, information managed by application servers can be organized to support such sharing as is allowed by the server, as well as advisory labeling.

2.4 MYSEOS

MYSEOS (depicted in Figure 2) is built on OpenBSD as a set of kernel enhancements to create labeled protection domains and a set of additional security services. The MYSEOS kernel associates security attributes with active and passive entities exported at the operating system interface. Enhancements include a protected security manager configured to interpret these attributes and enforce policy according to configuration-specific rules. An important policy for the MYSEOS kernel to enforce is that malicious code may neither exfiltrate confidentially-sensitive data nor corrupt information of higher integrity; to support this, the MYSEOS kernel provides multi-domain file system support, which provides for the global and persistent separation of data into its respective domains. Other security services that have been integrated into the MYSEOS kernel are described below.

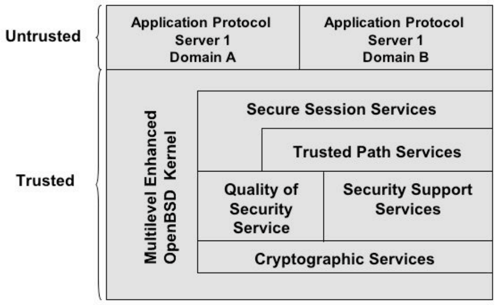
Untrusted
Trusted
Application Protocol Server 1 Domain A
Application Protocol Server 1 Domain B
Multilevel Enhanced OpenBSD Kernel
Secure Session Services
Trusted Path Services
Quality of Security Service
Security Support Services
Cryptographic Services

Figure 2. MYSEA server

2.5 Trusted path services

The Trusted Path Services component supports multiple locally attached terminals, as well as multiple remote MYSEA workstations. Trusted Path Services maintains the state of the user-to-MYSEA interaction, for example, a user may be logged in with default security attributes, but may not have started a session executing untrusted application code. Trusted Path Services provides an interface to the Security Support Services component to support identification and authentication, negotiation of domain or domain range, password modification, account creation and deletion, and user security attribute maintenance. Once a session has been established, the Trusted Path Services provides a distributed Session Status Database to the Secure Session Services component.

2.6 Secure session services

The Secure Session Services component is used to launch instances of untrusted constrained application protocol servers. It provides trusted policy-sensitive services, with functionality similar to that of classic *inetd* implementations and supports standard application protocol transmissions. The Secure Session Services accesses the Session Status Database, maintained by the Trusted Path component, to determine the security attributes to associate with each application protocol server.

This Session Status Database contains tuples that uniquely identify the user, the client workstation associated with the user, the status of the user session, the security attributes of the session, and other security relevant information. Through a session status communication mechanism, information in the Session Status Database can be provided to distributed multi-policy platforms, thus providing a single sign-on and session level capability.

2.7 Quality of Security Service support

MYSEA can be integrated with an external resource or QoS manager to provide a means of dynamically managing its security and performance characteristics. The MYSEA QoSS Manager is the external QoSS interface to MYSEA, and governs security and performance factors of the various MYSEA components, for example, which application protocol servers the client may interact with, and the cryptographic protection characteristics of the underlying communication channels. The QoSS security and connectivity database is managed by the QoSS manager on the MYSEA server, and is distributed to the Trusted Path Extensions, as needed.

The Quality of Security Service manager provides a user interface so that decision makers can request the overall security posture of the network. This interface provides the decision maker with a simple set of choices, hiding the underlying complexity of the quality of security service mechanisms [9].

2.8 Constrained application protocol servers

The secure session server provides instances of standard protocol servers for each client or for equivalence classes of clients. The Session Status Database, which is managed by the trusted path services component, but is readable by the secure session server, is used to assign security attributes to protocol servers launched on behalf of a requesting client. Thus, the protocol servers are associated with domains reflecting the granularity of the policy enforced by the underlying trusted operating system.

Protocol servers take two forms. The first form is a standard, policy-unaware protocol server, e.g. HTTP. These servers are restricted to accessing files and other objects associated only with the particular domain associated with the session. The second type of server is policy-aware, e.g., a file system [6], and is able to take advantage of certain security policy domain relations that permit limited modes of access to certain other domains (e.g., "read down" for mandatory confidentiality policies).

Among the application servers we have adapted to the MYSEA environment are: Internet Mail Access Protocol (IMAP) based on the University of Washington IMAP server [4], Hypertext Transfer Protocol (HTTP) based on the Apache server [1], and Simple Mail Transfer Protocol (SMTP) based upon sendmail [3]. Each server required little or no code modification to be adapted to the multilevel environment. With a proper configuration of the policy-aware application protocol server, users can view information at or below their current session levels.

2.9 MYSEA workstations

MYSEA workstations consist of two physical components: a Trusted Path Extension and an untrusted personal computer (see Figure 3). The PCs are typical COTS products hosting a popular commercial operating system and a commercial application suite. The application suite contains client software intended to access standard application protocol servers. For example, mail service clients might include: Lotus Notes, Outlook, Pine, Postal, and Netscape [5]. A typical browser supports the client interface to web pages.

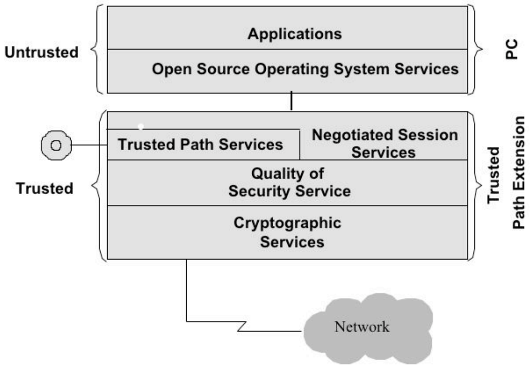

Figure 3. MYSEA Workstation

To ensure that object reuse requirements are met, workstations are managed to be, in effect, "diskless," with sufficient volatile RAM-disk capability to support a wide variety of user applications. The Trusted Path Extension satisfies object reuse requirements by ensuring that RAM and other volatile primary and secondary storage are purged with each change of session level or new user login at the workstation.

2.10 Trusted path extension

The trusted elements of the MYSEA system provide the locus of security policy enforcement. Not only do these elements provide runtime policy enforcement, but they must also provide services for the enforcement of supporting policies. To create a distributed TCB, the architecture includes a Trusted Path Extension at each workstation.

The Trusted Path Extension maintains its own self-protecting domain that is separate from the user and workstation domains. The use of a separate processor for the Trusted Path Extension ensures that it cannot be subverted by malicious software on the workstation. Architecturally, the Trusted Path Extension provides the PC's only access to the network.

The Trusted Path Extension has two form factors: an internal PCI card (planned for future development) and an external hand-held computer (per the current MYSEA prototype). In the PCI card format the Trusted Path Extension presents a NIC interface to the workstation. User trusted path I/O, including the secure attention key, is achieved via strictly controlled access to the PC keyboard and display. In the handheld format, the Trusted Path Extension performs IP network address translation for all IP traffic going between the PC and the LAN -- and user trusted path I/O occurs via the handheld's native keyboard and screen.

Simplicity has been a primary design goal for the Trusted Path Extension. The objective was not to construct a second operating system for the PC; it does not require the complexity and rich set of services provided by a typical PC (e.g. file system, printers and other peripheral drivers). The Trusted Path Extension can be viewed as a minimized embedded system that maintains no state of its own; instead, it functions as a "drone" in response to commands from the MYSEA server for controlling the workstation and managing I/O with the user. The Trusted Path Extension, under direction from the MYSEA server, supports the following services:

- Secure Attention Key – this service permits users to initiate unambiguous communication with MYSEOS for unspoofable presentation and capture of security critical data at the user interface. The secure attention key must cause a state change in the Trusted Path Extension such that an unforgeable communications path (viz. a *trusted path*) to MYSEOS is established.
- Trusted Path Services – when the trusted path is invoked, the user may elect to input security critical information, such as a password. The trusted path services ensure that prompts from the

server are displayed and that an input mechanism for replies is available.

- Controlled LAN Access – provide non-bypassable, controlled access to the LAN from the PC. Malicious software on the PC cannot bypass the Trusted Path.
- Communications and cryptographic services – provide protected communication channels between the server and the Trusted Path Extension. These protected communications are based upon protocols that support both the establishment and maintenance of a trusted path and session-level communications, such as to initiate communication with the server (via the secure attention key), as well as to receive and to respond to commands from the MYSEA Server.
- Negotiated Session Services – these mechanisms ensure trusted *object reuse* at the client PC for both primary and secondary storage. When a user chooses to change domains, certain policies require that information associated with the previous domain be purged from the untrusted PC, e.g. previous session information cannot be reused by subsequent sessions in conflict with the distributed security policy. The Trusted Path Extension ensures that object reuse requirements are met with each session change and as dictated by policy for session level changes. The Trusted Path Extension supports object reuse directives issued by MYSEOS. These directives may include both functional and procedural actions at the workstation.
- Control of Security Critical Activities –control the client and its resources at the time of boot and control security critical actions throughout the client session.
- Quality of Security Service - as networks become more complex and adaptive, it may be necessary to provide "security on demand." When conditions on the network change, requirements for security may also change. In response to a change notification, quality of security service mechanisms located on the Trusted Path Extension can modify the protection services afforded an ongoing session. The selection of protection mechanisms for communications between the client and the server may be based upon network conditions such as INFOCON mode. A version of IPSec adapted to provide automated, dynamic Quality of Security Service through the use of an enhanced version of a policy server such as Keynote [2] permits selection of protection mechanisms for MYSEA Servers.

3. Conclusion

We have presented the Monterey Security Enhanced Architecture (MYSEA), which provides a trusted distributed operating environment for enforcing multi-domain security policies, and which supports unmodified COTS productivity applications. The architecture encompasses a combination of many (untrusted) commercial components and relatively few trusted multi-domain components. Our prototype implementation utilizes a security-enhanced version of the OpenBSD operating system, called *MYSEOS,* as the policy enforcing *trusted computing base* (TCB). The architecture is general enough that it would easily accommodate a high assurance TCB, as well.

MYSEA introduces several innovations for protecting multiple data domains and for managing security policies and security services in support of critical applications, including:

- A distributed trusted architecture that utilizes commercial and open source applications to access multiple data domains.
- An open source trusted path mechanism.
- Techniques for vertical integration of security policy control functions with underlying security services.
- Single sign-on for access to a community of distributed multi-domain policy servers.

In the future, we plan several additions and enhancements to MYSEA. We have begun investigation of a ring mechanism [12] for open source operating systems, to help constrain the behavior of applications that run on MYSEOS and similar environments.

There are various systems and tools available to support the automated verification of computer system behavior. As a precursor to the formal analysis of the security behavior of MYSEA components, we have received support to perform a survey of available formal verification tools. That survey was started this summer.

We have recently started a project that includes the development of a very high assurance micro kernel. The goal for the Trusted Computing Exemplar Project is to provide a worked example of a high assurance system that can be used by the education community, government and industry. To further that aim, we plan to make the micro kernel, its development methodology and its evaluation evidence generally available through open source methods. As an early example of the application of the high assurance micro kernel, we plan to implement a high assurance Trusted Path Extension.

References

[1] Bersack, Evelyn, Implementation of a HTTP (Web) Server on a High Assurance Multilevel Secure Platform, Masters Thesis, Naval Postgraduate School, Monterey, California, December 2000.

[2] Blaze, Matt, Feigenbaum, Joan, and Keromytis, Angelos D., KeyNote: Trust Management for Public-Key Infrastructures, In Proceedings of the 1998 Security Protocols International Workshop, Springer LNCS vol. 1550, pp. 59 - 63. April 1998, Cambridge, England. Also AT&T Technical Report 98.11.1.

[3] Brown, Emma, SMPT on a High Assurance Mulitlevel Server, Masters Thesis, Naval Postgraduate School, Monterey, California, September 2000.

[4] Eads, Bradley, Developing a High Assurance Multilevel Mail Server, Masters Thesis, Naval Postgraduate School, Monterey, California, March 1999

[5] Everette, Theresa, Enhancement of Internet Message Access Protocol for User-Friendly Multilevel Mail Management, Masters Thesis, Naval Postgraduate School, Monterey, California, September 2000.

[6] Irvine, C. E., Acheson, T.B. and Thompson, M.F., "Building Trust into a Multilevel File System," Proc. of the 13th National Computer Security Conference, October 1990, Washington, DC

[7] Irvine, C. E., and Levin, T., "Quality of Security Service," in the Proceedings of the New Security Paradigms Workshop, September 2000.

[8] Irvine, C. E., Shifflett, D.J., Clark, P.C., Levin, T.E., Dinolt, G.W. "MYSEA Security Architecture," Naval Postgraduate School Technical Report, NPS-CS-02-006, May 2002.

[9] Mohan, Raj, Xml Based Adaptive Ipsec Policy Management In A Trust Management Context, Masters Thesis, Naval Postgraduate School, Monterey, California, September 2002

[10] National Computer Security Center, Computer Security Requirements Guidance For Applying The Department Of Defense Trusted Computer System Evaluation Criteria In Specific Environments, CSC-STD-004-85, June 1985.

[11] OpenBSD, “Audit Process,” http://www.openbsd.org/security.html#process, last modified December 2002, no expiration date set.

[12] Michael D. Schroeder, Jerome H. Saltzer: A Hardware Architecture for Implementing Protection Rings. CACM 15(3): 157-170 (1972).